

Description

SYSTEM FOR ACTIVELY UPDATING A CRYPTOGRAPHY MODULE IN A SECURITY GATEWAY AND RELATED METHOD

BACKGROUND OF INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to a system for updating a cryptography module and related method, and more particularly, to a system which can actively update a cryptography module in a security gateway and related method.

[0003] 2. Description of the Prior Art

[0004] The most popular security gateway in the today's market is the Virtual Private Network Gateway (VPN Gateway). Virtual Private Networks allow users to login to a public network, such as the Internet or an Asynchronous Transfer Mode (ATM) Network, from any terminal in the world. The environment of a Virtual Private Network is the same as the environment of a local network, such as intranet or

extranet. Therefore, Virtual Private Networks can offer the convenience of a public network and the safety of an internal network. Due to these advantages, authorized users can set up exclusive communications with other users, companies, branches, and customers by the Internet to transmit important information to each other. As shown in Fig. 1, the structure of a Virtual Private Network includes several user computer systems 10, 30, and 40 respectively having VPN gateways 104, 304, and 404, and setting up VPN tunnels 602 via the Internet 50 to transmit important information between each other. When a user uses one of the user computer systems 10, 30, or 40 to login to the internal computer system, such as a server 20, the computer systems 10, 30, or 40 can set up a VPN tunnel 602 by the respective VPN gateway 104, 304, or 404 for remote data access.

[0005] The technique of tunneling mentioned above involves using one of the three common communication protocols IPSEC, PPTP, and L2TP. In a public network, such as the Internet, a safe tunnel, the same as the one used in internal networks, is set up to protect packets of confidential information by encapsulation. This can prevent transmitted confidential information from being stolen by hackers.

Also, transmission of confidential information can be for other systems such as security qualification, ID authentication, and decryption/encryption. This makes VPN gateways safe and diverse.

[0006] Decryption/encryption in VPNs mentioned above is of two types: symmetric secret key cryptography and asymmetric public key cryptography. For example, in the IPSEC protocol, using an Internet Key Exchange (IKE) protocol having IKE phase 1 and phase 2 generates a public key to protect a secret key transmitted to the receiver so that the receiver can use the secret key to open the encrypted information. The purpose of IKE is to set up, identify, and exchange a security association (SA) for the identification of the transmitter and the receiver; establishing the decryption/encryption algorithm; and generating, exchanging, and setting up the key. The key length of the VPN, the decryption/encryption algorithm, and the decryption/encryption executing functions are recorded in a decryption/encryption module in each VPN gateway.

[0007] Although most manufacturers of VPN gateways provide their own designs with standard decryption/encryption as the decryption/encryption according to the IPSEC protocol mentioned above, when considering the safety, stability,

efficiency of execution, and communication through the entire system, the update of the decryption/encryption module is combined with the update of the kernel firmware of the VPN gateway. That is to say, when the decryption/encryption module needs to be updated, the kernel firmware must be updated at the same time. The prior art update is shown in Fig. 3. First, in step S200, a user computer system (such as the user computer system 10 in Fig. 1) is connected to a server (such as the server 20 in Fig. 1) through a browser and the Internet. Then, in step S210, a new kernel firmware is loaded into the storage device of the user computer system (storage device 102 in Fig. 1). In steps 220 and 230, the new kernel firmware is uploaded to a VPN gateway 104" via a Web GUI 114 shown in Fig. 2. In step 240, a kernel update module 126 of a current library 124 of the VPN gateway 104 (Fig. 2) is used to start updating a kernel 134 with the new kernel firmware. Subsequently, in step 250, the kernel update module 126 updates the kernel firmware, including updating an decryption/encryption module 128 of the current library 124. Finally, in step 260, the VPN gateway is rebooted to achieve step 270, completing the update of the new decryption/encryption module.

[0008] The prior art has the following drawbacks:

[0009] (1) Although each decryption/encryption module only occupies a part of program code in the VPN gateway, the safety of the decryption/encryption module is very important. However, the decryption/encryption module provided by each manufacturer cannot satisfy all the requirements of users. In the prior art, when VPN gateways are made, the original configuration setting permanently stores the decryption/encryption module in the current library of the VPN gateway. When users want to use other kinds of decryption/encryption modules, they have to update the entire kernel firmware of the machine. Therefore, manufacturers must provide many versions of kernel firmware decryption/encryption modules to meet these requirements. Time is expended to load the kernel firmware, reducing the efficiency and introducing room for error. In addition, the costs of maintaining the different versions can be quite significant.

[0010] (2) The prior art lacks a necessary function. That is, users of VPN gateways should be allowed to develop and set up decryption/encryption modules according to their requirements instead of using the standard module provided by manufacturers. If VPN gateway products can pro-

vide the function of allowing users to update or add decryption/encryption modules by themselves, such flexible design would increase potential customers and promote the expansion of decryption/encryption modules in VPN gateways.

SUMMARY OF INVENTION

[0011] It is therefore a primary objective of the claimed invention to provide a system for actively updating a cryptography module in a security gateway and related method. The claimed invention allows a user of the gateway to update the decryption/encryption modules of the extended library of the gateway through a module update unit instead of updating the decryption/encryption modules along with the entire kernel firmware. This can reduce the expended time, increase the efficiency of operation, and reduce the maintenance cost.

[0012] It is another objective of the claimed invention to provide a system for actively updating a cryptography module in a security gateway and related method. The claimed invention allows the user of the gateway to easily define the decryption/encryption modules and add the newly defined decryption/encryption modules into an extended library through a defined module unit and a module update unit.

This can simplify the update and promote the expansion of decryption/encryption modules of gateways so that network transmission is much safer.

[0013] It is another objective of the claimed invention to provide a system for actively updating a cryptography module in a security gateway and related method. The claimed invention allows the user of the gateway to easily select a decryption/encryption module in a window through a Web GUI for adding the new or updated decryption/encryption module into the extended library. This can increase the efficiency of system operation and the convenience of operation.

[0014] In order to achieve the objectives mentioned above, the system for actively updating a cryptography module in a security gateway of the present invention is set in a security gateway, such as the VPN gateway according to IPSEC, which includes a current library, a kernel, and a daemon, and is connected between at least one user computer system and a network system.

[0015] The system for actively updating a cryptography module in a security gateway of the present invention comprises a Web GUI, a module update unit, a defined module unit, an extended library, an extended library interface, and a con-

figuration set unit. The Web GUI can generate at least one window in the user computer system, the window having a decryption/encryption module update system to allow the user to selectively upload a new decryption/encryption module into the security gateway through the window. The module update unit is set in the current library to actively update a corresponding decryption/encryption module in the extended library according to the new decryption/encryption module uploaded to the security gateway or to add the new uploaded decryption/encryption module into the extended library. The extended library is used to accommodate decryption/encryption modules mentioned above. The extended library interface assists the extended library to communicate with the current library and the kernel. The configuration set unit is a system file for setting the execution process according to an IPSEC protocol. When a decryption/encryption module is updated or increased, the key exchange process is also updated.

[0016] The method of actively updating a cryptography module in a security gateway of the present invention is used in a security gateway. The security gateway is connected between at least one user computer system and a network

system. The method includes: using the browser of the user computer system through the network system connection to the website of a gateway manufacturer to download a new decryption/encryption module program code into the user computer system; rebooting a Web GUI of the security gateway to generate at least one window in the user computer system, the window having a decryption/encryption module update system; selecting an uploaded decryption/encryption module, such as a defined decryption/encryption module, from the window provided by the Web GUI; uploading the selected decryption/encryption module to the security gateway, a module update unit actively updating a corresponding decryption/encryption module in an extended library according to the uploaded decryption/encryption module or adding the new decryption/encryption module into the extended library; updating the key exchange process of the security gateway according to an IKE protocol; and rebooting the security gateway to execute the new key exchange process.

[0017] These and other objectives of the present invention will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the preferred embodiment that is illustrated in the various fig-

ures and drawings.

BRIEF DESCRIPTION OF DRAWINGS

- [0018] Fig. 1 is a schematic diagram of a security gateway applied in the structure of a network system.
- [0019] Fig. 2 is a diagram of a security gateway having decryption/encryption modules according to the prior art.
- [0020] Fig. 3 is a flowchart of updating decryption/encryption modules of the security gateway in Fig. 2.
- [0021] Fig. 4 is a diagram of a present invention system for actively updating a cryptography module in a security gateway.
- [0022] Fig. 5 is a flowchart of updating decryption/encryption modules of the security gateway according to the present invention.
- [0023] Fig. 6 is a flowchart of a key exchange process of the present invention according to an IKE protocol.

DETAILED DESCRIPTION

- [0024] Please refer to Fig. 4. Fig. 4 shows a system 110 of actively updating a cryptography module in a security gateway according to the present invention. The system 110 is set in a network security gateway 104 as shown in Fig. 1. The network security gateway 104 is a VPN gateway con-

nected to the Internet 50, the VPN gateway complying with an IPSEC protocol for allowing a user computer system 10 to set up a VPN tunnel 602 to securely transmit confidential information to other user computer systems 30 and 40. Additionally, the security gateway 104 includes a current library 124 having a default decryption/encryption module A, a kernel 164 being the operating system of the security gateway 104, and a daemon 174 for sequentially arranging processes of the entire gateway such as storing information, transmitting information, updating decryption/encryption modules, etc.

[0025] The system 110 includes a Web GUI, a module update unit 126, a defined module unit 128, an extended library 134, an extended library interface 144, and a configuration set unit 154. The extended library interface 144 generates at least one window having a plurality of decryption/encryption module update systems in the user computer system 10 to allow the user to easily operate or set the security gateway 104. Suppose that a system is to update a corresponding decryption/encryption module in the security gateway 104. Another system (defined as the decryption/encryption module system) allows the user to add and store an extra defined decryption/encryption module into

the security gateway 104. Of course, before the user starts the Web GUI 114 for updating the decryption/encryption modules of the security gateway 104, the connection is necessarily established from the Internet to the website of the gateway manufacturer. However, a key difference is that only downloading a new decryption/encryption program code module to the user computer system is required, instead of downloading the entire kernel firmware.

[0026] The module update unit 126 is set in the current library 124 of the security gateway 104 and actively updates or adds the decryption/encryption module into the extended library 134 according to the uploaded decryption/encryption module from the Web GUI 114. The extended library 134 includes a plurality of decryption/encryption modules such as an updated decryption/encryption module B and a defined decryption/encryption module C.

[0027] The defined module unit 128 is set in the current library 124 of the security gateway 104 and connects to the defined decryption/encryption module system of the Web GUI 114 to generate the window of the defined module unit 128 (the window is not shown here) to allow the user to fill in a field in the window with a description of the de-

defined decryption/encryption module. The description includes an algorithm, algorithmic identifier, data encryption block size, key length, and decryption/encryption executing function. The parameters of the decryption/encryption executing function include a data address, data block size, key information, key length, initial vector, and decryption/encryption flag, etc.

[0028] When the defined module unit 128 completes the defined decryption/encryption module C, the defined decryption/encryption module C must be uploaded by the Web GUI 114 to allow the module update unit 126 to add the defined decryption/encryption module C into the extended library 134. The extended library interface 144 assists the extended library 134 in communicating with the current library 124 and the kernel 164.

[0029] The configuration unit 154, such as a system file, is used to set up the execution process according to an IPSEC protocol. When a decryption/encryption module is updated or added, the current key exchange process according to an IKE protocol is also updated as the following steps: (1) determine if the current library 124 has a default decryption/encryption module in each IKE phase 1 or 2; (2) if no, further determine if the extended library 134

has any new or updated decryption/encryption module until selecting a decryption/encryption module for the key exchange process; and (3) after the IKE completes the key exchange process, inform the kernel 164 of an update to the security argument (SA) according to the IPSEC protocol.

[0030] Please refer to Fig. 5. Fig. 5 is an flowchart of updating decryption/encryption modules of the security gateway according to the present invention. The steps include:

[0031] Step S300: Connect a browser of the user computer system 10 to the server 20 of the security gateway manufacturer through the Internet 50.

[0032] Step S302: Download the new decryption/encryption module to the storing device 102 of the user computer system 10.

[0033] Step S304: Start the Web GUI of the security gateway 104.

[0034] Step S306: Select the uploaded decryption/encryption module from the window provided by the Web GUI 114. If selecting the defined decryption/encryption module C, go to step S308.

[0035] Step 308: Start a window of the defined module unit 128 for providing the user with an instruction to fill in a field in the window with a description of the defined decryp-

tion/encryption module. The description includes an algorithm, algorithmic identifier, data encryption block size, key length, and decryption/encryption executing function. The parameters of the decryption/encryption executing function include a data address, data block size, key information, key length, initial vector, and decryption/encryption flag, etc. After the user determines that the parameters of decryption/encryption module C are correct, go to step S310.

[0036] Step S310: Upload the defined decryption/encryption module C to the security gateway 104.

[0037] If the updated decryption/encryption module B is selected in step S304, step S310 uploads the updated decryption/encryption module B to the security gateway 104.

[0038] Step S312: The module update unit 126 of the security gateway 104 determines that the uploaded decryption/encryption module is a defined decryption/encryption module or an updated decryption/encryption module. If the result is the updated decryption/encryption module, go to step S316. Otherwise, if the result is the defined decryption/encryption module, go to step S314.

[0039] Step S314: Add the defined decryption/encryption module into the extended library 134.

- [0040] Step S316: Update the corresponding decryption/encryption module in the extended library 134.
- [0041] Step S317: Update the key exchange process in the configuration set unit 154 of the security gateway 104 according to an IKE protocol. (the key change process will be described later.)
- [0042] Step S318: Reboot the security gateway 104 so that the security gateway 104 can execute the updated key exchange process.
- [0043] Step 320: Complete the update of the decryption/encryption module.
- [0044] Please refer to Fig. 6. Fig. 6 is a flowchart of the updated key exchange process according to an IKE protocol in step S318. Fig. 6 applies in a previous communication of confidential transmission between a receiver and a transmitter (such as the user computer systems 10 and 30 in Fig. 1). The steps include:
- [0045] Step S400: Initiate the current IPSEC SA of the security gateway 104.
- [0046] Step S410: Execute an IKE phase 1.
- [0047] Step S420: Determine if the current library 124 has an appropriate decryption/encryption module, such as a default decryption/encryption module. If yes, go to step S430.

- [0048] Step S430: Select the key of the default decryption/en-
cryption module and the operation logic to communicate
with the receiver.
- [0049] If the current library 124 does not find any acceptable de-
cryption/encryption module in step S420, go to Step
S422: Further determine if the extended library 134 has
an appropriate decryption/encryption module, such as a
new or updated decryption/encryption module. If yes,
step S430 is processed. Then, step S430 selects the new
or updated decryption/encryption module to communi-
cate with the receiver.
- [0050] Step S440: Execute an IKE phase 2.
- [0051] Steps S450, S455, and S460 respectively repeat the ac-
tions of steps S420, S422 and S430. If step S422 or S455
does not find any appropriate decryption/encryption
module, then step S462 is processed.
- [0052] Step S462: The system generates an error message.
- [0053] Step S470: Complete all the key exchange processes of
IKE phases 1 and 2.
- [0054] Step S480: Inform the kernel 164 of the security gateway
104 of an update to the current SA according to IPSEC
protocol.

[0055] As mentioned above, the present invention allows the user of the gateway to simply update the decryption/encryption modules of the extended library of the gateway through a module update unit instead of updating the decryption/encryption modules along with the entire kernel firmware. This can reduce the setting time, increase the efficiency of operation, and reduce the maintenance cost. In addition, the present invention makes it convenient for the user to define the decryption/encryption modules through the defined module unit and the Web GUI. This promotes the expansion of decryption/encryption modules of security gateways.

[0056] Those skilled in the art will readily observe that numerous modifications and alterations of the device may be made while retaining the teachings of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.